```
ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE SCANM
OBJECT MODULE PLACED IN SCAN.OBJ
COMPILER INVOKED BY:  :F0:SCAN.PLM DEBUG OPTIMIZE(2) DATE(FEBRUARY 2, 1982) XREF


            $title ('SCANNER MODULE')
  1         scanm:
            do;

            /*

              modified  3/26/81  R. Silberstein
              modified  3/30/81  R. Silberstein
              modified  4/10/81  R. Silberstein
              modified  9/2/81   R. Silberstein

            */

            $include (:f1:macro.lit)
        =   $nolist
            $include (:f1:struc.lit)
        =   $nolist
            $include (:f1:equals.lit)
        =   $nolist
            $include (:f1:files.ext)
        =   $nolist
            $include (:f1:predef.ext)
        =   $nolist
            $include (:f1:subr2.ext)
        =   $nolist
            $include (:f1:print.ext)
        =   $nolist
            $include (:f1:global.ext)
        =   $nolist

                  /* Variables : */

 107  1     dcl
             eoffound       byte,   /* true if end-of-file is found */
             lowercase      byte,   /* false if stringinput, otherwise true */
             crfound        byte,   /* true if previous input was CR */
             printready     byte,   /* true if output line to be printed */
             stacksave      addr;   /* save of stack pointer */


            /* Routine to perform unnormal exit from module */

 108  1     exit: proc;
 109  2       stackptr=stacksave;   /* restore input stack */
 110  2     end exit;

            /* Put printcharacter into printfile output buffer */

 111  1     putprintchar: proc(ch);
 112  2       dcl ch byte;
 113  2       sourcebuf(sourceptr)=ch;
```

CP/M-86 v.1.1 (Vol. IV)
COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # C81-162

```
114  2      if sourceptr < last(sourcebuf) then$do
116  3        sourceptr=sourceptr+1;
117  3      end$if;
118  2    end putprintchar;


           /* Read single character from input file, Put characters
              except CR-LF to printbuffer, Convert to uppercase letters */

119  1    read$input: proc byte;
120  2      dcl ch byte;
121  2      if eoffound then call exit;  /* unnormal exit */
                                        /* read byte from file */
123  2      if include$on then ch=inincludebyte; else ch=insourcebyte;
126  2      if ch=end$of$file then$do   /* test for end-of-file */
128  3        eoffound=true;
129  3      else$do
131  3        if crfound and ch=lf then$do /* ignore LF after CR */
133  4          ch=space;
134  4        else$do;
137  4          if ch=cr then$do    /* test for CR */
139  5            crfound=true;
140  5          else$do;
143  5            crfound=false;
144  5            call putprintchar(ch);
145  5            if ch=lf then ch=space; /* interpret LF within line as space */
               end$if;
148  4        end$if;
149  3      end$if;
150  2      if not lowercase then$do   /* convert to uppercase */
152  3        ch=upper(ch);
153  3      end$if;
154  2      return ch;
155  2    end read$input;


           /* skip blanks and tab's in input */

156  1    skip$blanks: proc;
157  2      do while nextch=space or nextch=tab;
158  3        nextch=read$input;
159  3      end$while;
160  2    end skip$blanks;


           /* Put character into accumulator */

161  1    putaccum: proc(ch);
162  2      dcl ch byte;
163  2      accum(acclen)=ch;
164  2      if acclen < last(accum) then$do
166  3        acclen=acclen+1;
167  3      end$if;
168  2    end put$accum;


           /* Routine to scan remainder of token until a non-
```

```
                alphanumeric character is found, Skip blanks
                behind token */

169   1     get$remainder: proc(numb);
170   2       dcl (cont,numb) byte;
171   2       cont=true;
172   2       do while cont;
173   3         do while alphanumeric(nextch:=read$input);
174   4           call putaccum(nextch);
175   4         end$while;
176   3         cont=false;
177   3         if nextch = '@' or nextch = '_' then$do
179   4           cont=true;
180   4           if numb then call putaccum(nextch);
                end$if;
              end$while;
184   2       call skipblanks;
185   2     end get$remainder;

              /* Routine to scan a text string, Called from SCAN */

186   1     stringr: proc;
187   2       dcl cont byte;
188   2       lowercase=true;
189   2       acclen=0;
190   2       cont=true;
191   2       do while cont;
192   3         nextch=readinput;
193   3         do while nextch <> '''' and nextch <> cr;
194   4           call putaccum(nextch);
195   4           nextch=read$input;
196   4         end$while;
197   3         if nextch='''' then$do
199   4           if (nextch:=readinput) = '''' then$do /* interpret '' as ' */
201   5             call putaccum(nextch);
202   5           else$do
204   5             lowercase=false;
205   5             call skipblanks;
206   5             token.type=string;
207   5             cont=false;
208   5           end$if;
209   4         else$do
211   4           lowercase=false;
212   4           token.type=error;
213   4           cont=false;
214   4         end$if;
215   3       end$while;
216   2     end stringr;

              /* Routine to scan a number, Called from SCAN, Test syntax
                 of number, compute binary value, */

217   1     numbr: proc;
218   2       dcl
              nobase        byte, /* number system, 2,8,10 or 16 */
              maxlgth       byte, /* max legal no of digits */
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                (i,j)           byte, /* counters */
                ch              byte,
                value           addr, /* 16 bit binary value */
                errorflag       byte; /* syntax error flag */

219   2         errorflag=false;
220   2         call getremainder(true);   /* get rest of token */
221   2         ch=accum(acclen-1);   /* pick up last character of token */
222   2         j=acclen-2;

                /* B (binary) */
223   2         IF CH = 'B' THEN
224   2           do; nobase=2; maxlgth=16; end;

                /* O or Q (octal) */
228   2         ELSE IF CH = 'O' OR CH = 'Q' THEN
229   2           do; nobase=8; maxlgth=6; end;

                /* H (hexadecimal) */
233   2         ELSE IF CH = 'H' THEN
234   2           do; nobase=16; maxlgth=4; end;

                /* D (decimal) */
238   2         ELSE IF CH = 'D' THEN
239   2           do; nobase=10; maxlgth=5; end;

                /* no subscript, default=decimal */
                ELSE
243   2           do; nobase=10; maxlgth=5; j=j+1; end;

248   2         i=0ffh;       /* skip leading zeros */
249   2         do while accum(i:=i+1) = '0'; end;
251   2         if j < maxlgth+i then$do
253   3           value=0;      /* syntax check number, compute binary value */
254   3           do while i <= j;
255   4             ch=accum(i);
256   4             ch=ch-'0';
257   4             if ch > 9 then ch=ch-7;
259   4             if ch >= nobase then$do
261   5               errorflag=true;
262   5             end$if;
263   4             value=value*nobase+ch;
264   4             i=i+1;
265   4           end$while;
266   3         else$do
268   3           errorflag=true;
269   3         end$if;

270   2         if errorflag then$do
272   3           token.type=error;
273   3         else$do
275   3           token.type=number;
276   3           token.descr=0;
277   3           token.value=value;
278   3         end$if;

279   2       end numbr;
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                /* Routine to scan an identifier. Lookup identifier in table
                   for predefined symbols */

280   1         identr: proc;
281   2           call get$remainder(false);  /* get rest of token into accumulator */
                                              /* look up identifier */
282   2           if not pfind(acclen,.accum(0),.token) then$do
284   3             token.type=ident;
285   3           end$if;
286   2         end identr;

                      /* PUBLIC subroutines : */

287   1         scaninit: proc public;
288   2           eofset,eoffound,crfound,lowercase,printready=false;
289   2           CALL FILL (SPACE, SIZE (PREFIX), .PREFIX);
290   2           CALL FILL (SPACE, LENGTH (ABSADDR), .ABSADDR);
291   2           sourceptr,prefixptr=0;
292   2           call printinit;   /* initiate print module */
293   2           call rewindsource;
294   2           nextch=space;
295   2         end scaninit;

296   1         scan: proc public;

297   2           stacksave=stackptr;
298   2           if printready then$do
300   3             call print$source$line;
301   3             print$ready=false;
302   3           end$if;
303   2           call skipblanks;
304   2           if eoffound then$do
306   3             token.type=spec;
307   3             if crfound then$do
309   4               eoffound=false;
310   4               eofset=true;
311   4             else$do
313   4               printready=true;  /* terminate line before EOF */
314   4               crfound=true;
315   4               accum(0)=cr;
316   4             end$if;
317   3           else$do
319   3             acclen=1;
320   3             accum(0)=nextch;

                    /* identifier */
321   3             IF LETTER (NEXTCH) THEN call identr;

                    /* number */
323   3             ELSE IF DIGIT (NEXTCH) THEN call numbr;

                    /* string */
325   3             ELSE IF NEXTCH = '''' THEN call stringr;

                    /* special letter */
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                ELSE
327   3           do;
328   4           token.type=spec;
329   4           if nextch='!' then accum(0) = cr;
331   4           IF NEXTCH = ';' THEN$DO
333   5             DO WHILE ACCUM (0) <> CR;
334   6               ACCUM (0) = READINPUT;
335   6             END$WHILE;
336   5           END$IF;
337   4           nextch=space;
338   4           if crfound then$do
340   5             print$ready=true;
341   5           else$do
343   5             call skipblanks;
344   5           end$if;
345   4           end;

346   3         end$if;
347   2       end scan;

348   1       skip$rest$of$line: proc public;
349   2         do while accum(0) <> cr;
350   3           call scan;
351   3         end$while;
352   2       end skip$rest$of$line;

353   1       specialtoken: proc(tok) byte public;
354   2         dcl tok byte;
355   2         if (token.type=spec) and (accum(0)=tok) then return true;
357   2         return false;
358   2       end specialtoken;

359   1       skip$until: proc(tok) byte public;
360   2         dcl tok byte;
361   2         do forever;
362   3           if token.type=spec then$do
364   4             if accum(0)=tok then$do
366   5               call scan;
367   5               return true;
368   5             end$if;
369   4             if accum(0)=cr then return false;
              end$if;
372   3           call scan;
373   3         end$forever;
374   2       end skip$until;

375   1       emptyline: proc byte public;
376   2         return specialtoken(cr);
377   2       end emptyline;

378   1       end$module scanm;
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
49	0000H	2	A.	WORD PARAMETER 50
104	0000H	4	ABSADDR.	BYTE ARRAY(4) EXTERNAL(70) 290
104	0000H	1	ACCLEN	BYTE EXTERNAL(74) 163 164 166 189 221 222 282 319
104	0000H	1	ACCLENSAVE	BYTE EXTERNAL(77)
104	0000H	80	ACCUM.	BYTE ARRAY(80) EXTERNAL(75) 163 164 221 249 255 282 315 320 330 333 334 349 355 364 369
104	0000H	80	ACCUMSAVE.	BYTE ARRAY(80) EXTERNAL(76)
2			ADDR	LITERALLY 45 53 59 77 80 83 86 104 107 218
67	0000H		ALPHANUMERIC . . .	PROCEDURE BYTE EXTERNAL(24) STACK=0000H 173
70	0000H		ASCIICHAR.	PROCEDURE BYTE EXTERNAL(25) STACK=0000H
104	0003H	1	BASEINDEX.	BYTE MEMBER(OPERANDS)
104	0008H	1	BASEINDEX.	BYTE MEMBER(CURRENTSYMBOL)
3			BASEREGBIT	LITERALLY
3			BASEREGCOUNT . . .	LITERALLY
3			BREGBIT.	LITERALLY
3			BREGCOUNT.	LITERALLY
5			BYT.	LITERALLY
9	0000H	1	CH	BYTE PARAMETER 10
218	000FH	1	CH	BYTE 221 223 228 233 238 255 256 257 258 259 263
120	0008H	1	CH	BYTE 124 125 126 131 133 137 144 145 146 152 154
88	0000H	1	CH	BYTE PARAMETER 89
111	0004H	1	CH	BYTE PARAMETER AUTOMATIC 112 113
70	0000H	1	CH.	BYTE PARAMETER 71
73	0000H	1	CH	BYTE PARAMETER 74
161	0004H	1	CH	BYTE PARAMETER AUTOMATIC 162 163
67	0000H	1	CH	BYTE PARAMETER 68
64	0000H	1	CH	BYTE PARAMETER 65
61	0000H	1	CH	BYTE PARAMETER 62
15	0000H	1	CH	BYTE PARAMETER 16
58	0000H	1	CH	BYTE PARAMETER 59
12	0000H	1	CH	BYTE PARAMETER 13
104	0000H	2	CIP.	WORD EXTERNAL(39)
38	0000H		CLOSEHEX	PROCEDURE EXTERNAL(13) STACK=0000H
36	0000H		CLOSEINCLUDE . . .	PROCEDURE EXTERNAL(12) STACK=0000H
40	0000H		CLOSEPRINT	PROCEDURE EXTERNAL(14) STACK=0000H
32	0000H		CLOSESOURCE. . . .	PROCEDURE EXTERNAL(10) STACK=0000H
42	0000H		CLOSESYMBOL. . . .	PROCEDURE EXTERNAL(15) STACK=0000H
4			CODE	LITERALLY
104	0000H	1	CODEMACROFLAG. . .	BYTE EXTERNAL(96)
3			CODEMACROHEAD. . .	LITERALLY
104	0000H	2	CODEMACROPTR . . .	WORD EXTERNAL(91)
170	0009H	1	CONT	BYTE 171 172 176 179
187	000AH	1	CONT	BYTE 190 191 207 213
2			CR	LITERALLY 137 193 315 330 333 349 369 376
107	0006H	1	CRFOUND.	BYTE 131 139 143 288 307 314 338
104	0000H	2	CSCIP.	WORD EXTERNAL(44)
104	0000H	1	CSEGSPEC	BYTE EXTERNAL(42)
104	0000H	1	CSEGTYPE	BYTE EXTERNAL(40)
104	0000H	2	CSEGVALUE.	WORD EXTERNAL(41)

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
104  0000H    1  CSPEC. . . . . .  BYTE EXTERNAL(51)
104  0000H    2  CURCSEG. . . . . .  WORD EXTERNAL(47)
104  0000H    2  CURDSEG. . . . . .  WORD EXTERNAL(48)
104  0000H    2  CURESEG. . . . . .  WORD EXTERNAL(50)
104  0000H    9  CURRENTSYMBOL. . .  STRUCTURE EXTERNAL(37)
104  0000H    2  CURSSEG. . . . . .  WORD EXTERNAL(49)
 76  0000H    2  D. . . . . .  WORD PARAMETER  77
 79  0000H    2  D. . . . . .  WORD PARAMETER  80
 85  0000H    2  D. . . . . .  WORD PARAMETER  86
 82  0000H    2  D. . . . . .  WORD PARAMETER  83
  2            DCL. . . . . .  LITERALLY
 85  0000H       DECOUT . . . . . .  PROCEDURE EXTERNAL(30) STACK=0000H
104  0000H    1  DEFAULTDRIVE . . .  BYTE EXTERNAL(94)
  4            DELETEDSYMB. . . .  LITERALLY
104  0001H    1  DESCR. . . . . .  BYTE MEMBER(TOKEN)  276
 44  0000H    1  DEV. . . . . .  BYTE PARAMETER  45
 61  0000H       DIGIT. . . . . .  PROCEDURE BYTE EXTERNAL(22) STACK=0000H  323
  4            DOUBLEDEFINED. . .  LITERALLY
104  0000H    2  DSCIP. . . . . .  WORD EXTERNAL(46)
104  0000H    1  DSPEC. . . . . .  BYTE EXTERNAL(52)
  5            DWRD . . . . . .  LITERALLY
102  0000H       EJECT. . . . . .  PROCEDURE EXTERNAL(37) STACK=0000H
  2            ELSEDO . . . . . .  LITERALLY
375  045FH   11  EMPTYLINE. . . .  PROCEDURE BYTE PUBLIC STACK=0008H
  2            ENDCASE. . . . . .  LITERALLY
  2            ENDDO. . . . . .  LITERALLY
  2            ENDFOREVER . . . .  LITERALLY
  2            ENDIF. . . . . .  LITERALLY  146  181  370
  2            ENDMODULE. . . . .  LITERALLY
  2            ENDOFFILE. . . . .  LITERALLY  126
  2            ENDPROC. . . . . .  LITERALLY
  2            ENDWHILE . . . . .  LITERALLY  182
107  0004H    1  EOFFOUND . . . . .  BYTE  121  128  288  304  309
104  0000H    1  EOFSET . . . . . .  BYTE EXTERNAL(78)  288  310
 76  0000H       EQUAL. . . . . .  PROCEDURE BYTE EXTERNAL(27) STACK=0000H
  4            ERROR. . . . . .  LITERALLY  212  272
213  0010H    1  ERRORFLAG. . . . .  BYTE  219  261  268  270
104  0000H    1  ERRORPRINTED . . .  BYTE EXTERNAL(81)
104  0000H    2  ERRORS . . . . . .  WORD EXTERNAL(71)
104  0000H    2  ESCIP. . . . . .  WORD EXTERNAL(43)
104  0000H    1  ESPEC. . . . . .  BYTE EXTERNAL(54)
108  0000H   10  EXIT . . . . . .  PROCEDURE STACK=0002H  122
  2            FALSE. . . . . .  LITERALLY  143  176  204  207  211  213  219  281  288  301  309
                                    357  370
 55  0000H    2  FCBADR . . . . . .  WORD PARAMETER  56
 55  0000H       FILEABORT. . . . .  PROCEDURE EXTERNAL(20) STACK=0000H
  3            FILEISTRUCTURE . .  LITERALLY
  3            FILEOSTRUCTURE . .  LITERALLY
 47  0000H       FILESETUP. . . . .  PROCEDURE BYTE EXTERNAL(17) STACK=0000H
 58  0000H       FILL . . . . . .  PROCEDURE EXTERNAL(21) STACK=0000H  289  290
 44  0000H    2  FILNAM . . . . . .  WORD PARAMETER  45
 44  0000H    2  FILTYP . . . . . .  WORD PARAMETER  45
104  0000H    2  FIRSTMACROPTR. . .  WORD EXTERNAL(82)
  2            FOREVER. . . . . .  LITERALLY  361
  2            FORMFEED . . . . .  LITERALLY
104  0000H    1  FULLSYMBTAB. . . .  BYTE EXTERNAL(84)
169  00E8H   35  GETREMAINDER . . .  PROCEDURE STACK=0012H  220  281
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. =

```
105  0000H       GLOBALINIT . . . .  PROCEDURE EXTERNAL(97) STACK=0000H
104  0000H    5  HELP . . . . . .   BYTE ARRAY(5) EXTERNAL(92)
 77  0000H       HEX1OUT. . . . . .  PROCEDURE EXTERNAL(28) STACK=0000H
 82  0000H       HEX2OUT. . . . . .  PROCEDURE EXTERNAL(29) STACK=0000H
104  0000H    1  I. . . . . .       BYTE EXTERNAL(93)
218  000DH    1  I. . . . . .       BYTE      248  249  251  254  255  264
  4               IDENT. . . . . .  LITERALLY      284
280  02DDH   37  IDENTR . . . . . .  PROCEDURE STACK=0016H        322
 44  0000H       IFILESETUP . . . .  PROCEDURE EXTERNAL(16) STACK=0000H
104  0000H    1  IFLEVEL. . . . . .  BYTE EXTERNAL(86)
104  0000H    1  IFLIST . . . . . .  BYTE EXTERNAL(57)
104  0000H    1  INCLUDEDEFAULT . .  BYTE EXTERNAL(95)
104  0000H    1  INCLUDEON. . . . .  BYTE EXTERNAL(85)        123
  3               INDEXREGBIT. . . .  LITERALLY
  3               INDEXREGCOUNT. . .  LITERALLY
 20  0000H       ININCLUDEBYTE. . .  PROCEDURE BYTE EXTERNAL(4) STACK=0000H       124
  2               INIT . . . . . .  LITERALLY
 18  0000H       INSOURCEBYTE . . .  PROCEDURE BYTE EXTERNAL(3) STACK=0000H       125
104  0000H    1  INTELHEXON . . . .  BYTE EXTERNAL(79)
  3               IREGBIT. . . . . .  LITERALLY
  3               IREGCOUNT. . . . .  LITERALLY
218  000EH    1  J. . . . . .       BYTE      222  246  251  254
  4               LAB. . . . . .    LITERALLY
                  LAST . . . . . .  BUILTIN        114  164
  8               LEFTBRACKET. . . .  LITERALLY
                  LENGTH . . . . . .  BUILTIN        290
104  0000H    2  LENGTH . . . . . .  WORD MEMBER(OPERANDS)
104  0000H    2  LENGTH . . . . . .  WORD MEMBER(CURRENTSYMBOL)
 64  0000H       LETTER . . . . . .  PROCEDURE BYTE EXTERNAL(23) STACK=0000H      321
  2               LF . . . . . .    LITERALLY      131  145
  2               LIT. . . . . .    LITERALLY       2    3    4    5    6    7    8
107  0005H    1  LOWERCASE. . . . .  BYTE      150  188  204  211  288
104  0000H    2  MACROPTR . . . . .  WORD EXTERNAL(83)
104  0000H    1  MAXCOL . . . . . .  BYTE EXTERNAL(58)
218  000CH    1  MAXLGTH. . . . . .  BYTE      226  231  236  241  245  251
 76  0000H    1  N. . . . . .       BYTE PARAMETER       77
 58  0000H    1  N. . . . . .       BYTE PARAMETER       59
 49  0000H    1  N. . . . . .       BYTE PARAMETER       50
 85  0000H    2  N. . . . . .       WORD PARAMETER       86
 82  0000H    2  N. . . . . .       WORD PARAMETER       83
 79  0000H    1  N. . . . . .       BYTE PARAMETER       80
  4               NEGLECTED. . . . .  LITERALLY
104  0000H    1  NEXTCH . . . . . .  BYTE EXTERNAL(73)        157  158  173  174  177  181  192  193  194
                                      195  197  199  201  294  320  321  323  325  329  331  337
  5               NIL. . . . . .    LITERALLY
218  000BH    1  NOBASE . . . . . .  BYTE      225  230  235  240  244  259  263
104  0000H    1  NOERROR. . . . . .  BYTE EXTERNAL(80)
104  0000H    1  NOOPER . . . . . .  BYTE EXTERNAL(89)
  3               NOOVERCOUNT. . . .  LITERALLY
  3               NOOVERRIDEBIT. . .  LITERALLY
169  0004H    1  NUMB . . . . . .   BYTE PARAMETER AUTOMATIC      170  180
  4               NUMBER . . . . . .  LITERALLY      275
217  013BH  290  NUMBR. . . . . .   PROCEDURE STACK=0016H        324
  8               OAND . . . . . .  LITERALLY
  8               OEQ. . . . . .    LITERALLY
104  0006H    2  OFFSET . . . . . .  WORD MEMBER(OPERANDS)
104  0006H    2  OFFSET . . . . . .  WORD MEMBER(CURRENTSYMBOL)
```

```
  8                OGE. . . . . .   LITERALLY
  8                OGT. . . . . .   LITERALLY
  8                OLAST. . . . . .   LITERALLY
  8                OLE. . . . . .   LITERALLY
  8                OLENGTH. . . . . .   LITERALLY
  8                OLT. . . . . .   LITERALLY
  8                OMOD . . . . . .   LITERALLY
  8                ONE. . . . . .   LITERALLY
  8                ONOT . . . . . .   LITERALLY
  8                OOFFSET. . . . . .   LITERALLY
  8                OOR. . . . . .   LITERALLY
 26  0000H         OPENHEX. . . . . .   PROCEDURE EXTERNAL(7) STACK=0000H
 24  0000H         OPENINCLUDE. . . .   PROCEDURE EXTERNAL(6) STACK=0000H
 28  0000H         OPENPRINT. . . . .   PROCEDURE EXTERNAL(8) STACK=0000H
 22  0000H         OPENSOURCE . . . .   PROCEDURE EXTERNAL(5) STACK=0000H
 30  0000H         OPENSYMBOL . . . .   PROCEDURE EXTERNAL(9) STACK=0000H
104  0000H    33   OPERANDS . . . . .   STRUCTURE ARRAY(4) EXTERNAL(90)
  3                OPERANDSTRUC . . .   LITERALLY       104
  4                OPERATOR . . . . .   LITERALLY
  8                OPTR . . . . . .   LITERALLY
  8                OSEG . . . . . .   LITERALLY
  8                OSHL . . . . . .   LITERALLY
  8                OSHORT . . . . . .   LITERALLY
  8                OSHR . . . . . .   LITERALLY
  8                OTYPE. . . . . .   LITERALLY
  9  0000H         OUTHEXBYTE . . . .   PROCEDURE EXTERNAL(0) STACK=0000H
 12  0000H         OUTPRINTBYTE . . .   PROCEDURE EXTERNAL(1) STACK=0000H
 15  0000H         OUTSYMBOLBYTE. . .   PROCEDURE EXTERNAL(2) STACK=0000H
 52  0000H         OUTTEXT. . . . . .   PROCEDURE EXTERNAL(19) STACK=0000H
  8                OXOR . . . . . .   LITERALLY
104  0000H     1   PAGESIZE . . . . .   BYTE EXTERNAL(64
104  0000H     1   PASS . . . . . .   BYTE EXTERNAL(38)
  7                PCODEMACRO . . . .   LITERALLY
  7                PCSEG. . . . . .   LITERALLY
  7                PDB. . . . . .   LITERALLY
  7                PDBIT. . . . . .   LITERALLY
  7                PDD. . . . . .   LITERALLY
  7                PDSEG. . . . . .   LITERALLY
  7                PDW. . . . . .   LITERALLY
  7                PEJECT . . . . . .   LITERALLY
  7                PEND . . . . . .   LITERALLY
  7                PENDIF . . . . . .   LITERALLY
  7                PENDM. . . . . .   LITERALLY
  7                PEQU . . . . . .   LITERALLY
  7                PESEG. . . . . .   LITERALLY
 49  0000H         PFIND. . . . . .   PROCEDURE BYTE EXTERNAL(18) STACK=0000H      282
  7                PIF. . . . . .   LITERALLY
  7                PIFLIST. . . . . .   LITERALLY
  7                PINCLUDE . . . . .   LITERALLY
  7                PLIST. . . . . .   LITERALLY
  7                PMODRM . . . . . .   LITERALLY
  7                PNOIFLIST. . . . .   LITERALLY
  7                PNOLIST. . . . . .   LITERALLY
  7                PNOSEGFIX. . . . .   LITERALLY
  7                PORG . . . . . .   LITERALLY
  7                PPAGESIZE. . . . .   LITERALLY
  7                PPAGEWIDTH . . . .   LITERALLY
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
  7                 PRB. . . . . .   LITERALLY
104  0000H   240    PREFIX . . . . .  BYTE ARRAY(240) EXTERNAL(68)        289
104  0000H     1    PREFIXPTR. . . .  BYTE EXTERNAL(69)         291
  7                 PRELB. . . . . .  LITERALLY
  7                 PRELW. . . . . .  LITERALLY
 91  0000H         PRINTCRLF. . . .  PROCEDURE EXTERNAL(32) STACK=0000H
104  0000H     1   PRINTDEVICE. . .  BYTE EXTERNAL(61)
 97  0000H         PRINTINIT. . . .  PROCEDURE EXTERNAL(35) STACK=0000H       292
 93  0000H         PRINTNEWPAGE . .  PROCEDURE EXTERNAL(33) STACK=0000H
104  0000H     1   PRINTON. . . . .  BYTE EXTERNAL(55)
107  0007H     1   PRINTREADY . . .  BYTE       288  298  301  313  340
 88  0000H         PRINTSINGLEBYTE.  PROCEDURE EXTERNAL(31) STACK=0000H
 95  0000H         PRINTSOURCELINE.  PROCEDURE EXTERNAL(34) STACK=0000H       300
104  0000H     1   PRINTSWITCHOFF .  BYTE EXTERNAL(56)
 99  0000H         PRINTTERMINATE .  PROCEDURE EXTERNAL(36) STACK=0000H
  2                 PROC . . . . . .  LITERALLY        9   12   15   18   20   22   24   26   28   30   32
                                        34   36   38   40   42   44   47   49   67   70   79   82   85   88
                                        91   93   95   97   99  102  108  111  119  156  161  169  186  217
                                       280  287  296  348  353  359  375
  7                 PRS. . . . . .   LITERALLY
  7                 PRW. . . . . .   LITERALLY
  7                 PSEGFIX. . . . .  LITERALLY
  4                 PSEUDO . . . . .  LITERALLY
  7                 PSIMFORM . . . .  LITERALLY
  7                 PSSEG. . . . . .  LITERALLY
 58  0000H     2   PT . . . . . .   WORD PARAMETER       59
  7                 PTITLE . . . . .  LITERALLY
161  00C2H    29   PUTACCUM . . . .  PROCEDURE STACK=0004H         174  181  194  201
111  000AH    29   PUTPRINTCHAR . .  PROCEDURE STACK=0004H         144
  6                 RAH. . . . . .   LITERALLY
  6                 RAL. . . . . .   LITERALLY
  6                 RAX. . . . . .   LITERALLY
  6                 RBH. . . . . .   LITERALLY
  6                 RBL. . . . . .   LITERALLY
  6                 RBP. . . . . .   LITERALLY
  6                 RBX. . . . . .   LITERALLY
  6                 RCH. . . . . .   LITERALLY
  6                 RCL. . . . . .   LITERALLY
  6                 RCS. . . . . .   LITERALLY
  6                 RCX. . . . . .   LITERALLY
  6                 RDH. . . . . .   LITERALLY
  6                 RDI. . . . . .   LITERALLY
  6                 RDL. . . . . .   LITERALLY
  6                 RDS. . . . . .   LITERALLY
  6                 RDX. . . . . .   LITERALLY
119  0027H   121    READINPUT. . . .  PROCEDURE BYTE STACK=0008H         158  173  192  195  199  334
  2                 REENT. . . . . .  LITERALLY
  4                 REG. . . . . .   LITERALLY
  6                 RES. . . . . .   LITERALLY
 34  0000H         REWINDSOURCE . .  PROCEDURE EXTERNAL(11) STACK=0000H       293
  8                 RIGHTBRACKET . .  LITERALLY
  6                 RSI. . . . . .   LITERALLY
  6                 RSP. . . . . .   LITERALLY
  6                 RSS. . . . . .   LITERALLY
 76  0000H     2   S. . . . . .     WORD PARAMETER       77
 49  0000H     2   S. . . . . .     WORD PARAMETER       50
104  0000H    12   SAVESOURCE . . .  BYTE ARRAY(12) EXTERNAL(60)
```

COPYRIGHT (C) 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
296  0348H  189  SCAN . . . . . .  PROCEDURE PUBLIC STACK=001AH        350  366  372
287  0302H   70  SCANINIT . . . . .  PROCEDURE PUBLIC STACK=000AH
  1  0000H       SCANM. . . . . .  PROCEDURE STACK=0000H
  3              SEGMBIT. . . . . .  LITERALLY
  3              SEGMCOUNT. . . . .  LITERALLY
104  0004H    2  SEGMENT. . . . . .  WORD MEMBER(OPERANDS)
104  0004H    2  SEGMENT. . . . . .  WORD MEMBER(CURRENTSYMBOL)
  3              SEGTYPEBIT . . . .  LITERALLY
  3              SEGTYPECOUNT . . .  LITERALLY
104  0003H    1  SFLAG. . . . . .  BYTE MEMBER(OPERANDS)
104  0003H    1  SFLAG. . . . . .  BYTE MEMBER(CURRENTSYMBOL)
104  0000H    1  SIMFORM. . . . . .  BYTE EXTERNAL(65)
               SIZE . . . . . .  BUILTIN        289
136  00A0H   41  SKIPBLANKS . . . .  PROCEDURE STACK=000CH        184  205  303  343
348  0405H   17  SKIPRESTOFLINE . .  PROCEDURE PUBLIC STACK=001EH
359  0432H   45  SKIPUNTIL. . . . .  PROCEDURE BYTE PUBLIC STACK=0020H
104  0000H   80  SOURCEBUF. . . . .  BYTE ARRAY(80) EXTERNAL(66)        113  114
104  0000H   12  SOURCENAME . . . .  BYTE ARRAY(12) EXTERNAL(59)
104  0000H    1  SOURCEPTR. . . . .  BYTE EXTERNAL(67)        113  114  116  291
  2              SPACE. . . . . .  LITERALLY      133  146  157  289  290  294  337
  4              SPEC . . . . . .  LITERALLY      306  328  355  362
353  0416H   28  SPECIALTOKEN . . .  PROCEDURE BYTE PUBLIC STACK=0004H       376
104  0000H    2  SSCIP. . . . . .  WORD EXTERNAL(45)
104  0000H    1  SSPEC. . . . . .  BYTE EXTERNAL(53)
               STACKPTR . . . . .  BUILTIN        109  297
107  0000H    2  STACKSAVE. . . . .  WORD      109  297
  4              STRING . . . . . .  LITERALLY      206
186  013BH  128  STRINGR. . . . . .  PROCEDURE STACK=0010H        326
  2              STRUC. . . . . .  LITERALLY      104
104  0002H    1  STYPE. . . . . .  BYTE MEMBER(OPERANDS)
104  0002H    1  STYPE. . . . . .  BYTE MEMBER(CURRENTSYMBOL)
  4              SYMBOL . . . . . .  LITERALLY
104  0000H    1  SYMBOLDEVICE . . .  BYTE EXTERNAL(62)
  3              SYMBOLHEAD . . . .  LITERALLY
  3              SYMBOLSTRUC. . . .  LITERALLY      104
104  0000H    2  SYMBTABADR . . . .  WORD EXTERNAL(88)
 52  0000H    2  T. . . . . .  WORD PARAMETER       53
  2              TAB. . . . . .  LITERALLY      157
 55  0000H    2  TEXTADR. . . . . .  WORD PARAMETER       56
  2              THENDO . . . . . .  LITERALLY      114  126  131  137  150  164  177  197  199  251  259
                                      270  282  298  304  307  331  338  362  364
104  0000H   30  TITLE. . . . . .  BYTE ARRAY(30) EXTERNAL(63)
359  0004H    1  TOK. . . . . .  BYTE PARAMETER AUTOMATIC      360  364
353  0004H    1  TOK. . . . . .  BYTE PARAMETER AUTOMATIC      354  355
104  0000H    4  TOKEN. . . . . .  STRUCTURE EXTERNAL(72)        206  212  272  275  276  277  282  284
                                      306  328  355  362
  2              TRUE . . . . . .  LITERALLY      128  139  171  179  188  190  220  261  268  310  313
                                      314  340  356  361  367
104  0000H    1  TYPE . . . . . .  BYTE MEMBER(TOKEN)       206  212  272  275  284  306  328  355  362
  3              TYPEBIT. . . . . .  LITERALLY
  3              TYPECOUNT. . . . .  LITERALLY
  4              UDEFSYMB . . . . .  LITERALLY
 73  0000H       UPPER. . . . . .  PROCEDURE BYTE EXTERNAL(26) STACK=0000H      152
 99  0000H    1  USEFACT. . . . . .  BYTE PARAMETER      100
218  0002H    2  VALUE. . . . . .  WORD      253  263  277
104  0002H    2  VALUE. . . . . .  WORD MEMBER(TOKEN)       277
  4              VARIABLE . . . . .  LITERALLY
```

COPYRIGHT © 1981, 1982
INTEL RESEARCH

```
     5            WRD. . . . . .   LITERALLY


MODULE INFORMATION:

     CODE AREA SIZE     = 046AH   1130D
     CONSTANT AREA SIZE = 0000H     0D
     VARIABLE AREA SIZE = 0011H    17D
     MAXIMUM STACK SIZE = 0020H    32D
     848 LINES READ
     0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____